APPEAL BRIEF UNDER 37 C.F.R. § 41.37

TABLE OF CONTENTS

Page

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellants:	Gregory Marlan et al.	Examiner: Ryan Dare
Serial No.:	10/631,988	Group Art Unit: 2186
Filed:	July 31, 2003	Docket No.: 3160.750US1
Customer No.:	21186	Confirmation No.: 2188
Title:	DETECTION AND CONTROL OF RESOURCE CONGESTION BY A NUMBER OF PROCESSORS	

APPEAL BRIEF UNDER 37 CFR § 41.37

Mail Stop Appeal Brief- Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

The Appeal Brief is presented in support of the Notice of Appeal to the Board of Patent Appeals and Interferences, filed on January 11, 2011, from the Final Rejection of claims 1-46 of the above-identified Application, as set forth in the Final Office Action dated November 12, 2010.

The Commissioner of Patents and Trademarks is hereby authorized to charge Deposit Account No. 19-0743 in the amount of $540.00 which represents the requisite fee set forth in 37 C.F.R. § 41.20(b)(2). Appellants respectfully request consideration and reversal of the Examiner's rejections of the pending claims.

1. REAL PARTY IN INTEREST

The real parties in interest of the above-captioned patent Application are SILICON GRAPHICS, INC., WELLS FARGO FOOTHILL CAPITAL, INC., GENERAL ELECTRIC CAPITAL CORPORATION, and MORGAN STANLEY & CO., INCORPORATED, as evidenced by the followings, respectively:

an assignment from Gregory Marlan et al. to SILICON GRAPHICS, INC., recorded April 12, 2004, at Reel 015200, Frame 0272-0276;

an assignment from SILICON GRAPHICS, INC. AND SILICON GRAPHICS FEDERAL, INC. to WELLS FARGO FOOTHILL CAPITAL, INC., recorded August 19, 2005, at Reel 016871, Frame 0809-0840;

an assignment from SILICON GRAPHICS, INC. to GENERAL ELECTRIC CAPITAL CORPORATION, recorded October 24, 2006, at Reel 018545, Frame 0777-0841; and

an assignment from GENERAL ELECTRIC CAPITAL CORPORATION to MORGAN STANLEY & CO., INCORPORATED, recorded October 18, 2007, at Reel 019995, Frame 0895-0960.

2. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to Appellants that will have a bearing on the Board's decision in the present appeal.

3. STATUS OF THE CLAIMS

The present Application was filed on July 31, 2003 with claims 1-40. In response to a Non-Final Office Action dated October 28, 2005, Appellants amended claims 27 and 34-40. In response to a Non-Final Office Action dated February 6, 2008, Appellants amended claims 1, 4, 5, 7-9, 11, 12, 14, 17, 19, 20, 24, 27, 28, 31, 32, 34, 35, 38, and 39. In response to a Non-Final Office Action dated October 7, 2008, Appellants amended claims 1-5, 8, 12, 17-22, 27, 30-34, and 37-40 and added claims 41-44. In response to a Final Office Action dated July 24, 2009, Appellants amended claims 1, 5, 8, 12, 17, 22, 31, and 38. In response to an Advisory Action dated December 30, 2009, Appellants filed a Request for Continued Examination and amended claims 1, 5, 8, 12, 17, 22, 27, 31, 34, and 38. In response to a Non-Final Office Action mailed on March 19, 2010, Appellants added claims 45 and 46. In response to a Final Office Action mailed on November 12, 2010 (hereinafter the "Office Action"), Appellants filed a Notice of Appeal on January 11, 2011. Claims 1-45 stand twice rejected, remain pending, and are the subject of the present Appeal.

4. STATUS OF AMENDMENTS

No amendments have been made subsequent to the Final Office Action dated November 12, 2010.

5. SUMMARY OF CLAIMED SUBJECT MATTER

This summary is presented in compliance with the requirements of Title 37 C.F.R. §41.37(c)(I)(v), mandating a "concise explanation of the subject matter defined in each of the independent claims involved in the appeal." Nothing contained in this summary is intended to change the specific language of the claims described, nor is the language of this summary to be construed to limit the scope of the claims in any way.

Specific paragraph and line numbers are merely exemplary and are given below merely as an aid in understanding various inventive subject matters presented. The page, paragraph, and line numbers relate to Appellant's as-filed Application wherein the line numbers are with respect to the page in which the lines are found.

Aspects of the present inventive subject matter of independent claims 1, 5, 8, 12, 17, 22, 27, 31, 34, 38, and 45 include, but are not limited to, a method, apparatus and computer-readable storage medium to detect and control resource congestion.

INDEPENDENT CLAIM 1

1. An apparatus comprising:

a load/store unit that includes a retry logic that is to retry access to a memory resource operatively coupled to the apparatus after receipt from the memory resource of a negative acknowledgment for an attempt to access the memory resource by the load/store unit (*see* FIG. 2, elements 218 & 286 and page 15, line 27 through page 16, line 6); and

a congestion detection logic to output a signal that indicates that the memory resource is congested based on receipt from the memory resource of a consecutive number of negative acknowledgments in response to access requests to the memory resource (*see* FIGS. 2 & 3, element 282 and page 16, line 24 through page 17, line 7).

INDEPENDENT CLAIM 5

5. A processor comprising:

a functional unit to attempt to access data from memory coupled to the processor based on an access request, wherein the functional unit is to retry attempts to access of the data based on other access requests after receipt from the memory of a negative acknowledgment in response to the attempt to access the data (*see* FIG. 2, elements 218 & 286 and page 15, line 27 through page 16, line 6); and

a congestion detection logic to detect congestion of access of the data from the memory based on receipt from the memory of a consecutive number of negative acknowledgments that exceed a threshold prior to access of the data (*see* FIGS. 2 & 3, element 282 and page 16, line 24 through page 17, line 7); and

a congestion control logic to disable the functional unit from the attempts to access the data from the memory for a time period after congestion is detected (*see* FIG. 2, element 284 and page 17, lines 7-22).

INDEPENDENT CLAIM 8

8. A processor comprising:

a functional unit to attempt to access a cache line in a cache memory coupled to the processor based on an access request (*see* FIG. 1, elements 102 & 104 and page 13, lines 4-26), wherein the functional unit is to retry attempts to access the cache line based on additional access requests after receipt from the cache memory of a negative acknowledgment in response to the attempt to access the data (*see* FIG. 2, elements 218 & 286 and page 15, line 27 through page 16, line 6);

a congestion detection logic to detect congestion of access of the cache line in the cache memory based on an average number of negative acknowledgments received from the cache memory that exceed a threshold prior to access of the data (*see* FIGS. 2 & 3, element 282, FIGS. 12 & 13 and page 29, lines 4-21); and

a congestion control logic to disable the functional unit from attempts to access the cache line in the cache memory for a time period after congestion is detected (*see* FIG. 2, element 284 and page 17, lines 7-22).

INDEPENDENT CLAIM 12

12. A system comprising:

a cache memory to store data (*see* FIG. 1, element 102 and page 12, lines 20-26); and

a first processor to attempt to access data from the cache memory based on access requests (*see* FIG. 1, element 104 and page 13, lines 4-26), wherein the first processor includes a congestion detection logic to detect congestion of access to the data from the cache memory based on receipt from the cache memory of a consecutive number of negative acknowledgments in response to the access requests (*see* FIGS. 2 & 3, element 282 and page 16, line 24 through page 17, line 7).

INDEPENDENT CLAIM 17

17. A system comprising:

a memory resource (*see* FIG. 1, element 102 and page 12, lines 20-26); and

a first processor having a load/store functional unit, the load/store functional unit to attempt to access the memory resource based on access requests (*see* FIG. 1, elements 102 & 104 and page 13, lines 4-26; *also see* FIG. 2, elements 218 & 286 and page 15, line 27 through page 16, line 6), wherein the first processor includes a congestion detection logic to detect congestion of access of the memory resource based on a consecutive number of negative acknowledgments received from the memory resource in response to the access requests prior to receipt received from the memory resource of a positive acknowledgment in response to one of the access requests within a first time period (*see* FIGS. 2 & 3, element 282 and page 16, line 24 through page 17, line 7; *see also* page 6, lines 1-7).

INDEPENDENT CLAIM 22

22. A system comprising:

a cache memory to include a number of cache lines for storage of data (*see* FIG. 1, element 102, page 4, lines 18-19 and page 12, lines 20-26); and

at least two processors, wherein a first processor of the at least two processors is to attempt to access the data in one of the number of cache lines in the cache memory based on access requests (*see* FIG. 1, element 104 and page 13, lines 4-26), wherein the first processor includes a congestion detection logic to detect congestion of access of a first cache line of the

number of cache lines in the cache memory based on a ratio of a number of negative acknowledgments to a number of positive acknowledgments received from the cache memory in response to the access requests (*see* FIGS. 2 & 3, element 282 and page 16, line 24 through page 17, line 7; *see also* FIG. 10 and page 25, line 26 through page 26, line 25).

INDEPENDENT CLAIM 27

27. A method comprising:

transmitting access requests, by a first processor, to access data in a memory (*see* FIG. 7, block 702 and page 20, lines 7-14);

receiving, by the first processor, a positive acknowledgment or a negative acknowledgment from a second processor that is associated with the memory based on one of the number of access requests (*see* FIG. 7, block 704 and page 20, lines 15-22); and

detecting congestion of the data in the memory based on receipt, by the first processor, of a consecutive number of negative acknowledgments from the second processor that exceed a first threshold, prior to receipt, by the first processor, of a positive acknowledgment from the second processor (*see* FIG. 7, blocks 708 & 710, and page 21, lines 1-21; *also see* FIG. 9, block 912 and page 24, line 25 through page 25, line 7).

INDEPENDENT CLAIM 31

31. A method comprising:

accessing, by at least one processor, a memory resource based on an access request (*see* FIG. 9, block 902 and page 23, lines 3-6);

receiving a positive acknowledgment if the memory resource is accessible (*see* FIG. 9, block 906 and page 23, lines 7-15; *see also* page 3, lines 3-9);

receiving a negative acknowledgment from the memory resource if the memory resource is not accessible (*see* FIG. 9, block 906 and page 23, lines 7-15; *see also* page 3, lines 3-9);

retrying accessing, by the at least one processor, of the memory resource based on a number of access requests (*see* FIG. 9, block 10 and page 23, line 20 through page 24, line 24; *see also* page 6, lines 25-27); and

detecting congestion of the memory resource based on receipt, by the at least one processor, from the memory resource of a consecutive number of negative acknowledgments that exceed a first threshold within a time period, prior to receiving a positive acknowledgment (*see* FIG. 9, block 910 and page 23, line 20 through page 24, line 24; *see also* page 6, line 27 through p. 7, line 2).

INDEPENDENT CLAIM 34

34. A computer storage medium that provides instructions, which when executed by a machine, cause said machine to perform operations comprising (*see* page 11, lines 1-10):

transmitting access requests, by a first processor, to access data in a memory (*see* FIG. 7, block 702 and page 20, lines 7-14);

receiving, by the first processor, a positive acknowledgment or a negative acknowledgment from a second processor that is associated with the memory based on one of the number of access requests (*see* FIG. 7, block 704 and page 20, lines 15-22); and

detecting congestion of the data in the memory based on receipt, by the first processor, of a consecutive number of negative acknowledgments from the second processor that exceed a first threshold, prior to receipt, by the first processor, of a positive acknowledgment from the second processor (*see* FIG. 7, blocks 708 & 710, and page 21, lines 1-21; *also see* FIG. 9, block 912 and page 24, line 25 through page 25, line 7).

INDEPENDENT CLAIM 38

38. A computer storage medium that provides instructions, which when executed by a machine, cause said machine to perform operations comprising (*see* page 11, lines 1-10):

accessing, by at least one processor, a memory resource based on an access request (*see* FIG. 9, block 902 and page 23, lines 3-6);

receiving a positive acknowledgment if the memory resource is accessible (*see* FIG. 9, block 906 and page 23, lines 7-15; *see also* page 3, lines 3-9);

receiving a negative acknowledgment from the memory resource if the memory resource is not accessible (*see* FIG. 9, block 906 and page 23, lines 7-15; *see also* page 3, lines 3-9);

retrying accessing, by the at least one processor, of the memory resource based on a number of access requests (*see* FIG. 9, block 10 and page 23, line 20 through page 24, line 24; *see also* page 6, lines 25-27); and

detecting congestion of the memory resource based on receipt, by the at least one processor, from the memory resource of a consecutive number of negative acknowledgments that exceed a first threshold within a time period, prior to receiving a positive acknowledgments (*see* FIG. 9, block 910 and page 23, line 20 through page 24, line 24; *see also* page 6, line 27 through p. 7, line 2).

INDEPENDENT CLAIM 45

45. A system comprising:

a network (*see* FIG. 1 and page 43, line 22).;

a plurality of processors connected to the network, wherein the plurality processors includes a first processor (*see* FIG. 1, element 104 and page 43, line 22); and

system memory, wherein the system memory includes remote memory that is across the network from the first processor, wherein the remote memory includes a cache memory to store data (*see* FIG. 1, element 102, page 4, lines 18-19 and page 12, lines 20-26);

wherein the first processor is configured to attempt to access the data from the cache memory based on access requests (*see* FIG. 1, elements 102 & 104 and page 13, lines 4-26) and wherein the first processor includes a congestion detection logic to detect congestion of access to the data from the cache memory based on receipt from the cache memory of a consecutive number of negative acknowledgments in response to the access requests (*see* FIGS. 2 & 3, element 282 and page 16, line 24 through page 17, line 7; *see also* page 6, lines 1-7).

DEPENDENT CLAIM 14

14. The system of claim 13, wherein the second processor is to transmit a negative acknowledgment back to the first processor through the hub controller if the data is not accessible, the second processor to transmit a positive acknowledgment back to the first processor through the hub controller if the data is accessible (*see* page 13, lines 4-5 & 15-26 and page 20, lines 9-14).

DEPENDENT CLAIM 19

19. The system of claim 18, wherein the second processor is to transmit a negative acknowledgment back to the first processor through the hub controller if the memory resource is not accessible, the second processor to transmit a positive acknowledgment back to the first processor through the hub controller if the memory resource is accessible (*see* page 13, lines 4-5 & 15-26 and page 20, lines 9-14).

DEPENDENT CLAIM 24

24. The system of claim 23, wherein the second processor is to transmit a negative acknowledgment back to the first processor through the hub controller if the one of the number of cache lines is not accessible, the second processor to transmit a positive acknowledgment back to the first processor through the hub controller if the one of the number of cache lines is accessible (*see* page 13, lines 4-5 & 15-26 and page 20, lines 9-14).

This summary does not provide an exhaustive or exclusive view of the present subject matter, and Appellant refers to each of the appended claims and its legal equivalents for a complete statement of the invention.

6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-46 are rejected under 35 U.S.C. § 103(a) as allegedly being obvious in view of Hughes et al. (U.S. Patent No. 6,427,193, hereinafter "Hughes"), Sachs et al. (U.S. Publication No. 2002/0009067, hereinafter "Sachs") and Ghose et al. (U.S. Publication No. 2002/0004842, hereinafter "Ghose").

7. ARGUMENT

A) The Applicable Law under 35 U.S.C. §103(a)

The Examiner has the burden under 35 U.S.C. § 103 to establish a *prima facie* case of obviousness. *In re Fine*, 837 F.2d 1071, 1074, 5 U.S.P.Q.2d (BNA) 1596, 1598 (Fed. Cir. 1988). As discussed in *KSR International Co. v. Teleflex Inc. et al.* (U.S. 2007), the determination of obviousness under 35 U.S.C. § 103 is a legal conclusion based on factual evidence. *See Princeton Biochemicals, Inc. v. Beckman Coulter, Inc.*, 7, 1336-37 (Fed. Cir. 2005). The legal conclusion, that a claim is obvious within § 103(a), depends on at least four underlying factual issues set forth in *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 17 (1966): (1) the scope and content of the prior art; (2) differences between the prior art and the claims at issue; (3) the level of ordinary skill in the pertinent art; and (4) evaluation of any relevant secondary considerations.

The U.S. Supreme Court decision of *KSR v. Teleflex* provided a tripartite test to evaluate obviousness.

> The rationale to support a conclusion that a claim would have been obvious is that ***all the claimed elements were known in the prior art*** and one skilled in the art could have combined the elements as claimed by known methods ***with no change in their respective functions***, and ***the combination would have yielded nothing more than predictable results*** to one of ordinary skill in the art.[1]

"If ***any of these [three] findings*** cannot be made, then this rationale [of combining prior art elements according to known methods to yield predictable results] cannot be used to support a conclusion that the claim would have been obvious." MPEP § 2143(A) (emphasis added).

Although other rationales for rejection under 35 U.S.C. §103(a) may exist, the basis for an obviousness rejection is still grounded in a consideration of all claim elements. "All words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970); *see also* MPEP § 2143.03.

---

[1] *See KSR*; *see also* MPEP § 2143 (emphasis added).

Additionally, to render the claimed subject matter obvious, the prior art references must teach or suggest every feature of the claims. *See* MPEP §§ 706.02(j) and 2143(A) (2008).

The *KSR* Court further held that "rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." *See In re Kahn*, 441 F. 3d 977, 988 (CA Fed. 2006) cited with approval in *KSR Int'l v. Teleflex Inc.*, 127 S. Ct. 1727, 1740-41 (2007).

Therefore, the Examiner must, as one of the inquiries pertinent to any obviousness inquiry under 35 U.S.C. §103, recognize and consider not only the similarities but also the critical differences between the claimed invention and the prior art. *See In re Bond, 910* F.2d 831,834, 15 USPQ2d 1566, 1568 (Fed. Cir. 1990), *reh'g denied,* 1990 U.S. App. LEXIS 19971 (Fed. Cir.1990). Critical differences in the prior art must be recognized (when attempting to combine references). *See id.* at 1568.

Moreover, the fact that a reference teaches away from a claimed invention is highly probative that the reference would not have rendered the claimed invention obvious to one of ordinary skill in the art. *See Stranco Inc. v. Atlantes Chemical Systems, Inc.,* 15 USPQ2d 1704, 1713 (Tex. 1990). When the prior art teaches away from combining certain known elements, discovery of a successful means of combining them is more likely to be nonobvious. *See Id.* at 4 citing *United States v. Adams,* 383 U.S. 39, 51-51 (1966).

"If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims prima facie obvious." *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959). These principles have not been changed by the ruling in *KSR*.

Finally, the Court in *KSR* also reaffirmed that "[a] factfinder should be aware, of course, of the distortion caused by hindsight bias and must be cautious of argument reliant upon ex post reasoning." *KSR Int'l Co. v. Teleflex Inc.*, 127 S. Ct. 1727, 82 USPQ2d at 1397; *see also Graham v. John Deere Co.,* 383 U.S. at 36, 148 USPQ at 474.

B) Discussion of the rejection of claims 1-46 under 35 U.S.C. § 103(a) as allegedly being obvious in view of Hughes, Sachs and Ghose.

On page 2 of the Final Office Action dated November 12, 2010, the Examiner rejected claims 1-46 under 35 U.S.C. § 103(a) as allegedly being obvious in view of Hughes, Sachs and Ghose. Since a proper *prima facie* case of obviousness has not been established in each case, Appellants respectfully traverse this rejection.

Independent claims 1, 5, and 8:

Claim 1 recites, in pertinent part:

> a load/store unit that includes a retry logic that is to retry access to a memory resource operatively coupled to the apparatus *after receipt from the memory resource of a negative acknowledgment for an attempt to access the memory resource* by the load/store unit; and
>
> a congestion detection logic *to output a signal that indicates that the memory resource is congested based on receipt from the memory resource of a consecutive number of negative acknowledgments* in response to access requests to the memory resource.

Elements similar to the above-quoted elements are recited in independent claims 5 and 8. Appellants respectfully submit that neither Hughes nor Sachs nor Ghose, alone or in combination, teaches or suggests the elements as taught by Appellants and claimed in claims 1, 5 and 8.

The Office Action states at p. 2, paragraph 4, lines 1-4, that col. 39, lines 20-32 of Hughes teaches the claimed elements, namely, "a load/store unit that includes a retry logic that is to retry access to a memory resource *after receipt from the memory resource of a negative acknowledgement for an attempt to access the memory resource* by the load/store unit." The Office Action further states at p. 3, lines 4-5, that "Hughes teaches [receiving] an NACK from the memory resource." Appellants respectfully disagree.

Hughes discusses a load/store unit that is configured to back off from retrying to access a cache line for which ownership was lost for a given time interval ("backoff time"). The load/store unit in Hughes automatically resumes attempting to reestablish the ownership for the cache line and to access the cache line after expiration of the given time interval ("backoff time"). This is described in Hughes, as follows:

> On the other hand, processor 10 loses sufficient ownership to complete a store memory operation if a cache line accessed by the store is previously in a state other than invalid or shared and the cache line is changed to invalid or shared in response to a snoop hit. In response to losing sufficient ownership, load/store unit 26 is configured to signal bus interface unit 37 to backoff (via a backoff signal on backoff line 336) and to increase the initial backoff time. The backoff time is increased each time the ownership is gained and then lost before the affected memory operation can be completed. Eventually, the memory operation may be completed (after the other processor successfully completes its memory operation) and the backoff time may be reset to its initial value. *The term "backoff time" refers to a time interval during which processors are configured to inhibit attempting to reestablish ownership of a cache line for which ownership was lost via a snoop operation*. It is noted that load/store unit 26 may be configured to perform the backoff internally (e.g. by not attempting to transmit commands to bus interface unit 37 for transmittal on the bus).

Hughes at col. 39, lines 16-35 (emphasis added).

> While processor 1 is backed off, processor 2 attempts to reestablish ownership of cache line A2 (reference numeral 374). Again, the operation may be a read with modify intent. Processor 1 loses ownership of cache line A1 in response to snooping processor 2's operation. Processor 2 then has exclusive ownership of both cache lines A1 and A2. However, prior to processor 2 completing the store operation, *processor 1's backoff interval expires and processor 1 attempts to reestablish ownership of cache line A1* (reference numeral 376). Processor 1 snoops the operation and loses ownership of cache line A1. Additionally, processor 1 enters a backoff interval of the initial length as well.

Hughes at col. 42, lines 17-43 (emphasis added); *see also* col. 42, lines 34-36.

As quoted above, the load/store unit in Hughes signals the bus interface unit 37 to back off from further attempting the cache line for which ownership was lost. Hughes does not teach or suggest that the load/store unit in Hughes receives a NACK from a memory resource including the cache line, as asserted in the Office Action. Furthermore, as noted above, the load/store unit in Hughes automatically resumes attempting to reestablish the ownership for the cache line after expiration of the backoff time, and does not operate to wait for the NACK from

the memory resource before resuming the attempt to reestablish the ownership. Given these teachings, Hughes fails to show "a load/store unit that includes a retry logic that is to retry access to a memory resource operatively coupled to the apparatus *after receipt from the memory resource of a negative acknowledgment for an attempt to access the memory resource* by the load/store unit," as recited in claim 1. Appellants are unable to find any such teaching within the bounds of Hughes, Sachs or Ghose, alone or in combination.

In addition, as admitted in the Office Action at p. 2, paragraph 4, line 5, Hughes fails to show "a congestion detection logic to output a signal that indicates that the memory resource is congested based on receipt from the memory resource of a consecutive number of negative acknowledgments in response to access requests to the memory resource," as recited in claim 1. However, the Office Action states at pages 2-3, paragraph 4, lines 5-12, that the combination of Sachs, Ghose and Hughes discloses these elements recited in claim 1. Appellants respectfully disagree.

Sachs discusses adapting to congestion in a network between a user equipment and a radio base station. *See* Sachs at paragraphs [0028], [0038] and [0044]. Under Sachs' approach, the negative acknowledgement is returned from the network, and not from a memory resource, as required by claims 1-46. In particular, Sachs teaches that "if a negative acknowledgment as an indication of congestion is returned from the network, the user equipment uses a longer "Subsequent Backoff Delay 2" to ease the load on the [network] channel." Sachs at paragraph [0052], lines 13-18. Furthermore, Sachs does not describe that the congestion in the network is caused by multiple retry requests to access the memory resource. Finally, as admitted in the Office Action at p. 3, lines 1-2, Sachs fails to teach that it takes more than one consecutive negative acknowledgements to indicate the network is congested. Therefore, Sachs fails to show "a congestion detection logic to output a signal that indicates that the memory resource is congested based on receipt from the memory resource of a consecutive number of negative acknowledgments in response to access requests to the memory resource," as recited in claim 1.

Ghose discusses coping with congestion in a network. *See* Ghose, paragraph [0117], lines 15-16. In particular, Ghose discusses "network congestion as evidenced by … the generation of a predetermined number of NACKs … during a predetermined time interval." Ghose, paragraph [0117], last six lines. That is, Ghose deals with the congestion in the network,

and not in a memory resource. Furthermore, Ghose does not describe that the congestion in the network is caused by multiple retry requests to access the memory resource. Finally, although Ghose uses a plurality of NACKs as an indication of the network congestion, Ghose fails to teach that the plurality of NACKs are consecutive, as required by claims 1-44.

The Office Action further states at p. 3, lines 4-5, that "Hughes teaches an NACK from the memory resource, so the combination of Hughes with Sachs teaches this limitation." As noted above, however, Hughes does not teach or suggest that the load/store unit in Hughes receives the NACK from the memory resource, as asserted in the Office Action. Therefore, the combination of Sachs, Ghose and Hughes fails to show "a congestion detection logic to output a signal that indicates that the memory resource is congested based on receipt from the memory resource of a consecutive number of negative acknowledgments in response to access requests to the memory resource," as recited in claim 1. Appellants are unable to find any such teaching within the bounds of Sachs, Ghose or Hughes, alone or in combination.

The arguments in support of the patentability of claim 1 similarly apply to independent claims 5 and 8, which each recite similar limitations.

Independent claims 12, 17, 22, 31, and 38:

Amended independent claim 12 recites, in pertinent part:

> a congestion detection logic to detect congestion of access to the data from the cache memory based on receipt from the cache memory of a consecutive number of negative acknowledgments in response to the access requests.

For the same reasons as have been noted with respect to independent claims 1, 5, and 8, Appellants respectfully submit that neither Hughes nor Sachs nor Ghose, alone or in combination, teaches or suggests the elements as taught by Appellants and claimed in independent claim 12. The arguments in support of the patentability of claim 12 similarly apply to independent claims 17, 22, 31, and 38, which each recite similar limitations.

Independent claims 27 and 34:

Independent claim 27 recites, in pertinent part:

> receiving, by the first processor, a positive acknowledgment or a negative acknowledgment from a second processor that is associated with the memory based on one of the number of access requests; and
>
> detecting congestion of the data in the memory based on receipt, by the first processor, of a consecutive number of negative acknowledgments from the second processor that exceed a first threshold, prior to receipt, by the first processor, of a positive acknowledgment from the second processor.

For reasons similar to those noted with respect to independent claims 1, 5, and 8, Appellants respectfully submits that neither Hughes nor Sachs nor Ghose, alone or in combination, teaches or suggests the above-quoted elements recited in independent claim 27.

The Office Action states that paragraph [0054] of Sachs teaches "receiving, by the first processor, a positive acknowledgement or a negative acknowledgement from a second processor that is associated with the memory based on the number of access requests." *See* the Office Action at p. 15, paragraph 36, lines 5-8. However, Sachs does not teach or suggest using the first processor that receives a positive or negative acknowledgement from the second processor, if any, associated with the memory to which the first processor transmits an access request, as asserted in the Office Action.

Furthermore, while the Office Action further states that "Hughes teaches a NACK from a second processor [associated] with memory" (Office Action at p. 15, paragraph 36, lines 15-16), the Office Action did not cite portions of Hughes that support the statement. Hughes does not teach or suggest using a NACK (transmitted) from a second processor, if any, associated with memory to which a first processor transmits an access request, as asserted in the Office Action.

For at least the reasons stated above, the combination of the cited documents does not teach or suggest "receiving, by the first processor, a positive acknowledgment or a negative acknowledgment *from a second processor* that is associated with the memory" or "detecting congestion of the data in the memory based on receipt, by the first processor, of a consecutive number of negative acknowledgments *from the second processor* that exceed a first threshold, prior to receipt, by the first processor, of a positive acknowledgment from the second processor,"

as taught by Appellants and claimed in independent claim 27. Appellants are unable to find any such teaching within the bounds of Hughes, Sachs or Ghose, alone or in combination.

The arguments in support of the patentability of claim 27 similarly apply to independent claim 34, which recites similar limitations.

Independent claim 45:

Independent claim 45 recites, in pertinent part:

> system memory, wherein the system memory includes remote memory that is across the network from the first processor, wherein the remote memory includes a cache memory to store data;
>
> wherein the first processor is configured to attempt to access the data from the cache memory based on access requests and wherein the first processor includes a congestion detection logic to detect congestion of access to the data from the cache memory based on receipt from the cache memory of a consecutive number of negative acknowledgments in response to the access requests.

The Office Action states at p. 21, paragraph 55, that "[c]laims 45-46 are rejected using similar logic." Appellants respectfully disagree.

For the same reasons as have been noted with respect to independent claims 1, 5, and 8, Appellants respectfully submit that neither Hughes nor Sachs nor Ghose, alone or in combination, teaches or suggests the claimed elements of independent claim 45, namely, "the first processor includes a congestion detection logic to detect congestion of access to the data from the cache memory based on receipt from the cache memory of a consecutive number of negative acknowledgments in response to the access requests." Nor does the combination of Hughes, Sachs and Ghose teach or suggest "the system memory includes remote memory that is across the network from the first processor, wherein the remote memory includes a cache memory to store data," as taught by Appellants and claimed in claim 45. Appellants are unable to find any such teaching within the bounds of Hughes, Sachs or Ghose, alone or in combination.

For at least the reasons stated above, Appellants respectfully submit that the Office Action fails to show a prima facie case of obviousness with respect to independent claims 1, 5, 8, 12, 17, 22, 27, 31, 34, 38, and 45, and that these claims are in condition for allowance. Therefore, reconsideration and allowance of independent claims 1, 5, 8, 12, 17, 22, 27, 31, 34, 38, and 45 are respectfully requested.

Dependent claims 14, 19, and 24:

In addition, claim 14 recites, in pertinent part,

> the second processor is to transmit a negative acknowledgment back to the first processor through the hub controller if the memory resource is not accessible, the second processor to transmit a positive acknowledgment back to the first processor through the hub controller if the memory resource is accessible.

Elements similar to the above-quoted elements are recited in claims 19 and 24. Neither Hughes nor Sachs nor Ghose, alone or in combination, teaches or suggests the elements as taught by Appellants and claimed in claims 14, 19, and 24.

The Office Action states at pages 9-10, paragraph 21, that FIG. 25 and col. 43, lines 15-25 of Hughes disclose these claimed elements. *See also* Office Action p. 12, paragraph 27 and p. 14, paragraph 33. Appellants respectfully disagree.

Hughes discusses using "bus bridge 202" coupled to two processors ("processor 10" and "processor 10a"). *See* Hughes at FIG. 25. Even if all requests between the processor 10 and the processor 10a are channeled through the bus bridge 202, as asserted in the Office Action at p. 10, lines 1-3, Hughes does not teach or suggest that the processor 10 attempts to access the "L2 cache 228a" that is locally coupled to the processor 10a. Furthermore, Hughes does not teach or suggest that the processor 10a transmits a positive acknowledgement or a negative acknowledgement to the processor 10 in response to receipt of the access request to the L2 cache 228a, if any, from the processor 10. Appellants are unable to find any such teaching as the above-quoted elements of claims 14, 19 and 24 within the bounds of Hughes, Sachs or Ghose, alone or in combination.

For at least the reasons stated above, Appellants respectfully submit that the Office Action fails to show a prima facie case of obviousness with respect to dependent claims 14, 19, and 24, and that these claims are in condition for allowance. Therefore, reconsideration and allowance of dependent claims 14, 19, and 24 are respectfully requested.

Claims 2-4, 6, 7, 9-11, 13-16, 18-21, 23-26, 28-30, 32, 33, 35-37, 39, and 40 are submitted to be allowable as depending from their respective independent claims 1, 5, 8, 12, 17, 22, 27, 31, 34 and 38, which are submitted to be allowable.

SUMMARY

For at least the reasons presented above, Appellants maintain that the rejections made under 35 U.S.C. §103(a) were improper. Appellants therefore respectfully submits that the claims are in condition for allowance, and that the cited art, alone or in combination, does not render the claims obvious. Appellants therefore request the Board to reverse the findings of the Examiner with regard to the rejections. Appellants further respectfully request allowance of the pending claims.

Respectfully submitted,

SCHWEGMAN, LUNDBERG & WOESSNER, P.A.
P.O. Box 2938
Minneapolis, MN 55402
(612) 371-2151

Date March 14, 2011 By /Hong, myunggon/

Myunggon Hong
Reg. No. 65,034

8. CLAIMS APPENDIX

1. An apparatus comprising:

a load/store unit that includes a retry logic that is to retry access to a memory resource operatively coupled to the apparatus after receipt from the memory resource of a negative acknowledgment for an attempt to access the memory resource by the load/store unit; and

a congestion detection logic to output a signal that indicates that the memory resource is congested based on receipt from the memory resource of a consecutive number of negative acknowledgments in response to access requests to the memory resource.

2. The apparatus of claim 1 further comprising a congestion control logic to disable the retry logic from retry accesses to the memory resource based on receipt of the signal from the congestion detection logic.

3. The apparatus of claim 2, wherein the congestion control logic is to exponentially increase the delay after the congestion detection logic is to detect congestion while the memory resource is currently congested.

4. The apparatus of claim 2, wherein the congestion control logic is to exponentially decrease the delay after the congestion detection logic receive a number of positive acknowledgments in response to access requests to the memory resource.

5. A processor comprising:

a functional unit to attempt to access data from memory coupled to the processor based on an access request, wherein the functional unit is to retry attempts to access of the data based on other access requests after receipt from the memory of a negative acknowledgment in response to the attempt to access the data; and

a congestion detection logic to detect congestion of access of the data from the memory based on receipt from the memory of a consecutive number of negative acknowledgments that exceed a threshold prior to access of the data; and

a congestion control logic to disable the functional unit from the attempts to access the data from the memory for a time period after congestion is detected.

6. The processor of claim 5, wherein the congestion control logic is to exponentially increase the time period after the congestion detection logic is to detect congestion while access to other data in the memory is congested.

7. The processor of claim 6, wherein the congestion control logic is to exponentially decrease the time period after the congestion detection logic receives a number of positive acknowledgments in response to attempts to access data in the memory.

8. A processor comprising:

a functional unit to attempt to access a cache line in a cache memory coupled to the processor based on an access request, wherein the functional unit is to retry attempts to access the cache line based on additional access requests after receipt from the cache memory of a negative acknowledgment in response to the attempt to access the data;

a congestion detection logic to detect congestion of access of the cache line in the cache memory based on an average number of negative acknowledgments received from the cache memory that exceed a threshold prior to access of the data; and

a congestion control logic to disable the functional unit from attempts to access the cache line in the cache memory for a time period after congestion is detected.

9. The processor of claim 8, wherein the average number of negative acknowledgments is within a window and wherein the congestion detection logic is to move the window over time of attempts to access the cache line by the functional unit.

10. The processor of claim 8, wherein the congestion control logic is to exponentially increase the time period after the congestion detection logic is to detect congestion while access of other cache lines in the cache memory is congested.

11. The processor of claim 8, wherein the congestion control logic is to exponentially decrease the time period after the congestion detection logic receives a number of positive acknowledgments in response to attempts to access other cache lines in the cache memory.

12. A system comprising:

a cache memory to store data; and

a first processor to attempt to access data from the cache memory based on access requests, wherein the first processor includes a congestion detection logic to detect congestion of access to the data from the cache memory based on receipt from the cache memory of a consecutive number of negative acknowledgments in response to the access requests.

13. The system of claim 12 further comprising:

a second processor associated with the cache memory;

a hub controller to receive the access requests from the first processor, the hub controller to forward the access requests to the second processor, wherein the second processor is to determine whether the data in the cache memory is accessible.

14. The system of claim 13, wherein the second processor is to transmit a negative acknowledgment back to the first processor through the hub controller if the data is not accessible, the second processor to transmit a positive acknowledgment back to the first processor through the hub controller if the data is accessible.

15. The system of claim 12, wherein the first processor further comprises a congestion control logic to disable the first processor from transmitting the access requests if the congestion detection logic determines that access to the data is congested.

16. The system of claim 12, wherein the congestion control logic is to disable the first processor from transmitting the access requests for a time period, wherein the time period is based on an exponential back off delay operation.

17. A system comprising:

a memory resource; and

a first processor having a load/store functional unit, the load/store functional unit to attempt to access the memory resource based on access requests, wherein the first processor includes a congestion detection logic to detect congestion of access of the memory resource based on a consecutive number of negative acknowledgments received from the memory resource in response to the access requests prior to receipt received from the memory resource of a positive acknowledgment in response to one of the access requests within a first time period.

18. The system of claim 17 further comprising:

a second processor associated with the memory resource;

a hub controller to receive the access requests from the first processor, the hub controller to forward the access requests to the second processor, wherein the second processor is to determine whether the memory resource is accessible.

19. The system of claim 18, wherein the second processor is to transmit a negative acknowledgment back to the first processor through the hub controller if the memory resource is not accessible, the second processor to transmit a positive acknowledgment back to the first processor through the hub controller if the memory resource is accessible.

20. The system of claim 17, wherein the first processor further comprises a congestion control logic to disable the load/store functional unit from attempting to access the memory resource if the congestion detection logic is to detect congestion of access of the memory resource.

21. The system of claim 17, wherein the congestion control logic is to disable the load/store unit from attempts to access the memory resource for a second time period, wherein the second time period is based on an exponential back off delay operation.

22. A system comprising:

a cache memory to include a number of cache lines for storage of data; and

at least two processors, wherein a first processor of the at least two processors is to attempt to access the data in one of the number of cache lines in the cache memory based on access requests, wherein the first processor includes a congestion detection logic to detect congestion of access of a first cache line of the number of cache lines in the cache memory based on a ratio of a number of negative acknowledgments to a number of positive acknowledgments received from the cache memory in response to the access requests.

23. The system of claim 22, wherein a second processor of the at least two processors is associated with the cache memory and wherein the system further comprises a hub controller, the hub controller to receive the access requests from the first processor, the hub controller to forward the access requests to the second processor, wherein the second processor is to determine whether the one of the number of cache lines is accessible.

24. The system of claim 23, wherein the second processor is to transmit a negative acknowledgment back to the first processor through the hub controller if the one of the number of cache lines is not accessible, the second processor to transmit a positive acknowledgment back to the first processor through the hub controller if the one of the number of cache lines is accessible.

25. The system of claim 22, wherein the first processor further comprises a congestion control logic to disable, for a time period, the first processor to attempt to access the data if the congestion detection logic is to detect congestion of access of the first cache line.

26. The system of claim 25, wherein the congestion control logic is to exponentially increase the time period after the congestion detection logic is to detect congestion while access to other cache lines in the cache memory.

27. A method comprising:

transmitting access requests, by a first processor, to access data in a memory;

receiving, by the first processor, a positive acknowledgment or a negative acknowledgment from a second processor that is associated with the memory based on one of the number of access requests; and

detecting congestion of the data in the memory based on receipt, by the first processor, of a consecutive number of negative acknowledgments from the second processor that exceed a first threshold, prior to receipt, by the first processor, of a positive acknowledgment from the second processor.

28. The method of claim 27 further comprising controlling access to the data in the memory if the consecutive number of negative acknowledgments, received by the first processor, exceeds the first threshold, prior to receipt of the positive acknowledgment.

29. The method of claim 28, wherein controlling access to the data in the memory comprises disabling transmitting of the access requests, by the first processor, for a time period.

30. The method of claim 29, wherein controlling access to the data in the memory comprises exponentially increasing the time period upon determining that the congestion is detected for other data in the memory while the time period has not expired.

31. A method comprising:

accessing, by at least one processor, a memory resource based on an access request;

receiving a positive acknowledgment if the memory resource is accessible;

receiving a negative acknowledgment from the memory resource if the memory resource is not accessible;

retrying accessing, by the at least one processor, of the memory resource based on a number of access requests; and

detecting congestion of the memory resource based on receipt, by the at least one processor, from the memory resource of a consecutive number of negative acknowledgments that exceed a first threshold within a time period, prior to receiving a positive acknowledgment.

32. The method of claim 31 further comprising controlling access to the memory resource if the consecutive number of negative acknowledgment, received by the at least one processor, exceeds the first threshold, prior to receipt of the positive acknowledgment.

33. The method of claim 31, wherein controlling access to the memory resource comprises disabling transmitting of the access requests, by the first processor, for a time period.

34. A computer storage medium that provides instructions, which when executed by a machine, cause said machine to perform operations comprising:

transmitting access requests, by a first processor, to access data in a memory;

receiving, by the first processor, a positive acknowledgment or a negative acknowledgment from a second processor that is associated with the memory based on one of the number of access requests; and

detecting congestion of the data in the memory based on receipt, by the first processor, of a consecutive number of negative acknowledgments from the second processor that exceed a first threshold, prior to receipt, by the first processor, of a positive acknowledgment from the second processor.

35. The computer storage medium of claim 34 further comprising controlling access to the data in the memory if the consecutive number of negative acknowledgments, received by the first processor, exceeds the first threshold, prior to receipt of the positive acknowledgment.

36. The computer storage medium of claim 35, wherein controlling access to the data in the memory comprises disabling transmitting of the access requests, by the first processor, for a time period.

37. The computer storage medium of claim 36, wherein controlling access to the data in the memory comprises exponentially increasing the time period upon determining that the congestion is detected for other data in the memory while the time period has not expired.

38. A computer storage medium that provides instructions, which when executed by a machine, cause said machine to perform operations comprising:

accessing, by at least one processor, a memory resource based on an access request;

receiving a positive acknowledgment if the memory resource is accessible;

receiving a negative acknowledgment from the memory resource if the memory resource is not accessible;

retrying accessing, by the at least one processor, of the memory resource based on a number of access requests; and

detecting congestion of the memory resource based on receipt, by the at least one processor, from the memory resource of a consecutive number of negative acknowledgments that exceed a first threshold within a time period, prior to receiving a positive acknowledgments.

39. The computer storage medium of claim 38 further comprising controlling access to the memory resource if the consecutive number of negative acknowledgments, received by the at least one processor, exceeds the first threshold, prior to receipt of the positive acknowledgment.

40. The computer storage medium of claim 39, wherein controlling access to the memory resource comprises disabling transmitting of the access requests, by the first processor, for a time period.

41. The apparatus of claim 1, wherein the memory resource comprises nonvolatile memory.

42. The system of claim 17, wherein the memory resource comprises nonvolatile memory.

43. The method of claim 31, wherein the memory resource comprises nonvolatile memory.

44. The computer storage medium of claim 38, wherein the memory resource comprises nonvolatile memory.

45. A system comprising:

a network;

a plurality of processors connected to the network, wherein the plurality processors includes a first processor; and

system memory, wherein the system memory includes remote memory that is across the network from the first processor, wherein the remote memory includes a cache memory to store data;

wherein the first processor is configured to attempt to access the data from the cache memory based on access requests and wherein the first processor includes a congestion detection logic to detect congestion of access to the data from the cache memory based on receipt from the cache memory of a consecutive number of negative acknowledgments in response to the access requests.

46. The system of claim 45, wherein the access requests comprise retry attempts by the first processor to access the data from the cache memory.

9. EVIDENCE APPENDIX

None.

10. RELATED PROCEEDINGS APPENDIX

None.